A1: C++ Starter

Re-submit Assignment

Due Aug 30 by 11:59pm **Points** 100 **Submitting** a file upload

Computer science is a wonderful tool for studying the world. Using computers, we can examine processes and phenomena that are difficult to study experimentally or through pure theory.

In this assignment, you will practice some basic C++ programming in the context of ecological simulation.

Background

Predator-prey relationships are a fundamental basis of ecological systems. Most ecologies are incredibly complex, but a few isolated systems have a more tractable formulation. Canadian fur-trapping companies studied fox-rabbit populations for decades, and found their relationship could be modeled by a few equations.

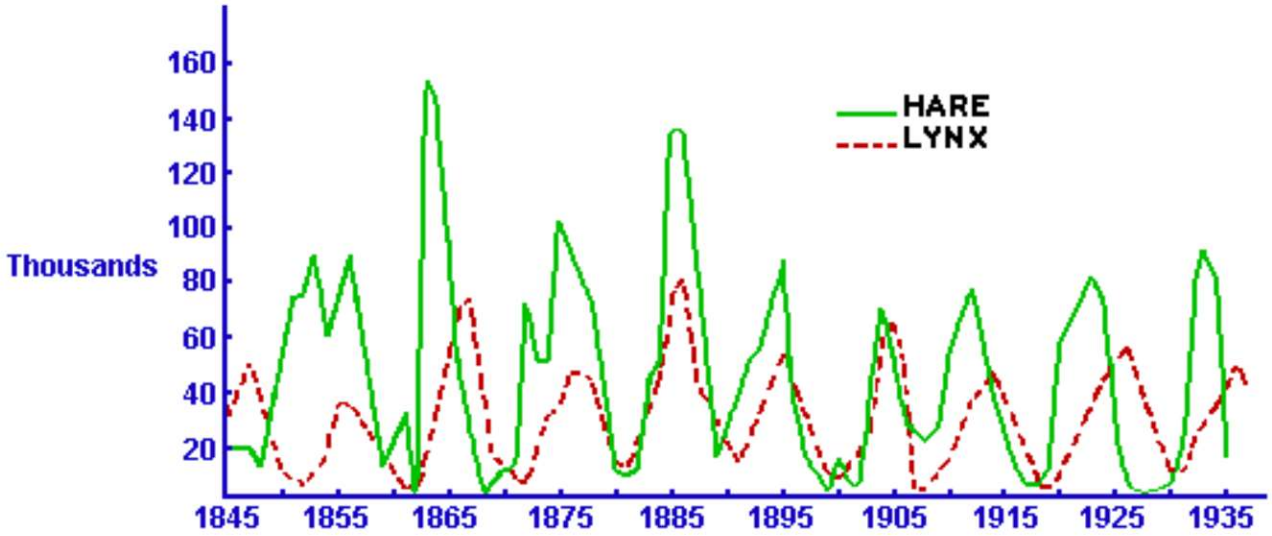

Figure 1: Source - http://www.math.duke.edu/education/ccp/materials/diffeq/predprey/pred1.html

The change in rabbit population depends on the natural rabbit growth rate, the carrying-capacity of the environment (basically, what the rabbits have to eat), and how often the rabbits get eaten by foxes. The change in fox population depends on their natural die-off rate and how much the population is boosted by having rabbits to eat.

The Lotka-Volterra with environmental carrying-capacity modification captures the above model. It has a number of parameters:

rabbitGrowth (g): the rate of increase of rabbits
predationRate (p): fraction of prey eaten per predator
foxPreyConversion (c): fraction of eaten prey turned into new predators
foxMortalityRate (m): a per capita mortality rate for foxes
carryCapacity (K): How many rabbits can be supported by the environment

The change in population with numRabbits (R) and numFoxes (F) is then:

deltaRabbit = gR(1-R/K) - pRF
deltaFoxes = cpRF - mF

Each iteration of the model computes the change in population and then updates R and F by adding deltaRabbit and deltaFoxes respectively.

Assignment

You will write a C++ program in a file EcoSim.cpp that computes the rabbit and fox populations and displays them as an ASCII chart, with each row showing the current rabbit and fox population. The specifics are:

1. A main function that sets the parameters needed for the update equation, that asks the user for an initial rabbit and fox population and that runs the simulation for 500 steps or until the predator or prey population goes below 1. Store the number of rabbits and foxes as doubles to let the equations work a little more smoothly. If the user enters anything that cannot be converted to a double for an initial population, give a warning message and terminate the program.
2. A population update function that takes in the model parameters and then the number of rabbits and number of foxes with a pass-by-reference style. Each iteration in the main function should call this update function. The function signature should be:

```
void updatePopulations(double g, double p, double c, double m, double K,
                       double& numRabbits, double& numFoxes);
```

3. A helper function plotCharacter that takes in an int number and a character and sends to std::cout number spaces followed by the character.
4. A charting function that returns void and takes in the number of rabbits and number of foxes and a fractional scale factor in that order. Using the plotCharacter function as a helper, it should draw a row of a text chart with an 'F' for foxes and 'r' for rabbits and '*' if the drawing of each would overlap. The characters should be positioned in the floor(num*scale) + 1 position from the left margin. So if the scale factor is 0.1 and the fox population is 100 and the rabbit population is 201.5, then the 11th character from the left should be an 'F' and the 21st character from the left should be an 'r'. If the fox population were 5, then the 'F' should be in the 1st space. Use a scale factor of 0.1 when testing this program. You may find it useful to use '-' during testing rather than ' ' to position the 'F' and 'r' characters. Use ' ' in your final submission.
5. A helper function incrementCounter that returns void and accepts a pointer to an integer. The function should add 1 to the value pointed to by the pointer. You must use this function to update your iteration count in the main function. This is purely make work to practice passing pointers.

A useful set of parameters to start testing is:

rabbitGrowth (g): 0.2
predationRate (p): 0.002
foxPreyConversion (c): 0.6
foxMortalityRate (m): 0.2
carryCapacity (K): 1000.0

Using those parameters, your chart should show an oscillation in fox-rabbit sizes. I used an initial population of 100 for each, although it seems pretty stable for other initial conditions.

Format

Neat formatting is required for your code. The file should have a header comment with your name, the assignment name and a sentence description of the program. Comment interesting or hard-to-understand portions of the code. Each function should have at least a 1 sentence comment describing what the function does (not what it is called). Use descriptive variable names, although the updatePopulations code can use the math shorthand parameter names I specified above.

Compiling

Your program should compile from the command line using the lab2 machines and the g++ compiler with a -std=c++11 flag set, which is our default for this course. The compilation should create an executable called EcoSim. Save the compilation command in a text file called compile.txt. Do not add notes or additional lines to the text file - we plan on executing it.

Submission

Submit to Canvas your EcoSim.cpp file and the compile.txt file.